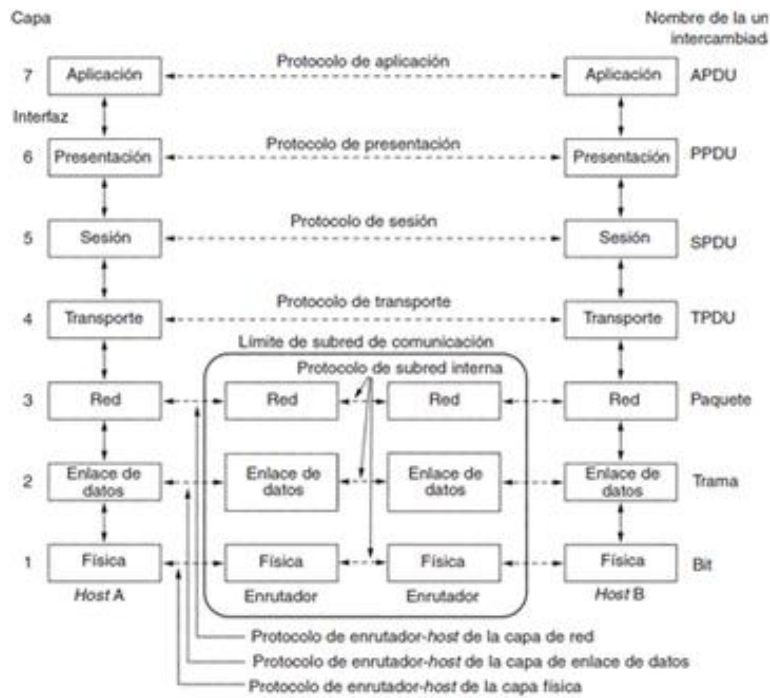


# Teoría de las Comunicaciones

*Segundo Cuatrimestre del 2014*

**Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires  
Argentina**



# Arquitectura de Redes

## Una revisión de los Fundamentos

# Múltiples Redes Globales

---

- ▶ BITNET, XEROX, DECNET ...
- ▶ ARPANET, CSNET, MILNET, UUCP ..Esta era la situación a mediados de los 80 , Quarterman realiza un survey de las principales redes Globales de la época [\*]
- ▶ Una arquitectura única de Red ( OSI –ISO)?
- ▶ **May 1983:** ISO publishes “ISO 7498:The Basic Reference Model for Open Systems Interconnection” as an international standard.

# Múltiples Redes Globales

UUCP		g!d.l!u	g!d.l!u	g!h.mailnet!u	
JUNET		u@d.l.arpa	u@d.l.arpa	u%h.mailnet@mit-multics.arpa	
From	Domains: To:	U: uk JANET	E: cdn, d/n, etc. EAN	COSAC	
ARPA Internet		u%d.U@cs.ucl.ac.uk	u%d.E%ubc.csnet@relay.cs.net	h/u%france.csnet@relay.cs.net	
CSNET Phonetnet		u%d.U@cs.ucl.ac.uk	u%d.E@ubc.csnet	h/u@france.csnet	
MAILNET		?	u%d.E@ubc.mailnet	?	
JANET		u@U.d	u@d.U	?	
EAN		u@d.U	u@d.E	h/u@france.csnet	
COSAC		?	?	h/u	
BITNET		u%d.U@ac.uk	u@d.E	h/u@france.csnet	
ACSNET		u%d.U@munnari.oz	u%d.E@munnari.oz	h/u%france.csnet@munnari.oz	
UUCP		g!cs.ucl.ac.uk!d.U!u	g!d.E!u	?	
JUNET		u@d.U.janet	u%d.E@ubc.csnet	h/u@france.csnet	
From	Domains: To:	R: A registry Xerox Internet	DEC's Easynet	IBM's VNET	
ARPA Internet		u.R@xerox.com	u%h.dec@decwrl.dec.com	u%h@ibm.com	
CSNET Phonetnet		u.R@xerox.com	u%h.dec@decwrl.dec.com	u%h@ibm.com	
MAILNET		?	?	?	
JANET		?	?	?	
EAN		u.R@xerox.com	u%h.dec@decwrl.dec.com	u%h@ibm.com	
COSAC		?	?	?	
BITNET		u.R@xerox.com	u%h.dec.com@decwrl.dec.com	u@vnet	
ACSNET		u.R%xerox.com@munnari.oz	u%h.dec.com@munnari.oz	u%h%ibm.com@munnari.oz	
UUCP		parcvax!u.R	decwrl!h.dec.com!u	g!ibm.com!u%h	
JUNET		u.R@xerox.com.arpa	u%h.dec@decwrl.dec.com.arpa	u%h@ibm.com.arpa	
From	Domains: To:	BITNET	A: oz.au ACSNET	UUCP	J: junet JUNET
ARPA Internet		u%h.bitnet@wiscvm.wisc.edu	u@d.A	u%h.uucp@g	u%d.J%utokyo-relay@relay.cs.net
CSNET Phonetnet		u%h.bitnet@relay.cs.net	u@d.A	u%h.uucp@g	u%d.J@utokyo-relay
MAILNET		?	u%d.A%g@mit-multics	?	u%d.J%csnet-relay@mit-multics
JANET		?	u%d.oz@uk.ac.ukc	?	u%d.J@uk.ac.ukc
EAN		u@h.bitnet	u@d.A	u@h.uucp	u%d.J@relay.cs.net
COSAC		adi/u%h.bitnet@relay.cs.net	?	adi/u%h.uucp	adi/u%h.J@relay.cs.net
BITNET		u@h	u%d.A@g	h1!h2!h!u@psuvax1	u%d.J@csnet-relay.csnet
ACSNET		u%h.bitnet@munnari.oz	u@d.A	u%h.uucp@munnari.oz	u%d.J@munnari.oz
UUCP		psuvax1!h.bitnet!u	seismo!munnari!d.A!u	h1!h2!h!u	g!d.l!u
JUNET		u@h.bitnet	u@d.A	u@h.uucp	u@d.J

Notes: From UUCP to CDNnet *ubc-ean* is a gateway; from EUnet to the European EAN networks there is one gateway per country; there is more than one gateway between BITNET and UUCP; UUCP, EUnet, and SDN are similarly addressed, so only one of them is listed here.

Abbreviations: u: user; h: host; g: gateway (unnamed here); d: domain.

Omissions: From company networks, with commercial networks, with the ARPA Internet.

# Como escribíamos una dirección .....

---

steve@cs.ucl.ac.uk  
... !ucl-cs!steve  
steve@uk.ac.ucl.cs

gb/bt/des/steve(ucl/cs)  
/C=GB/ADMD=BT/PRMD=DES/O=UCL/  
OU=CS/S=Kille/  
<C=gb;A=bt;P=des;O=ucl;S=steve;OU=cs>  
steve!ucl!cs&des%bt&gb  
steve!ucl!cs#des&bt.gb

steve@cs.ucl.des.bt.gb  
/C=/ADMD=/PRMD=UK/DD.=cs.ucl.ac/  
DD.=steve/

Via ARPA Internet  
Via UUCP  
Via JANET

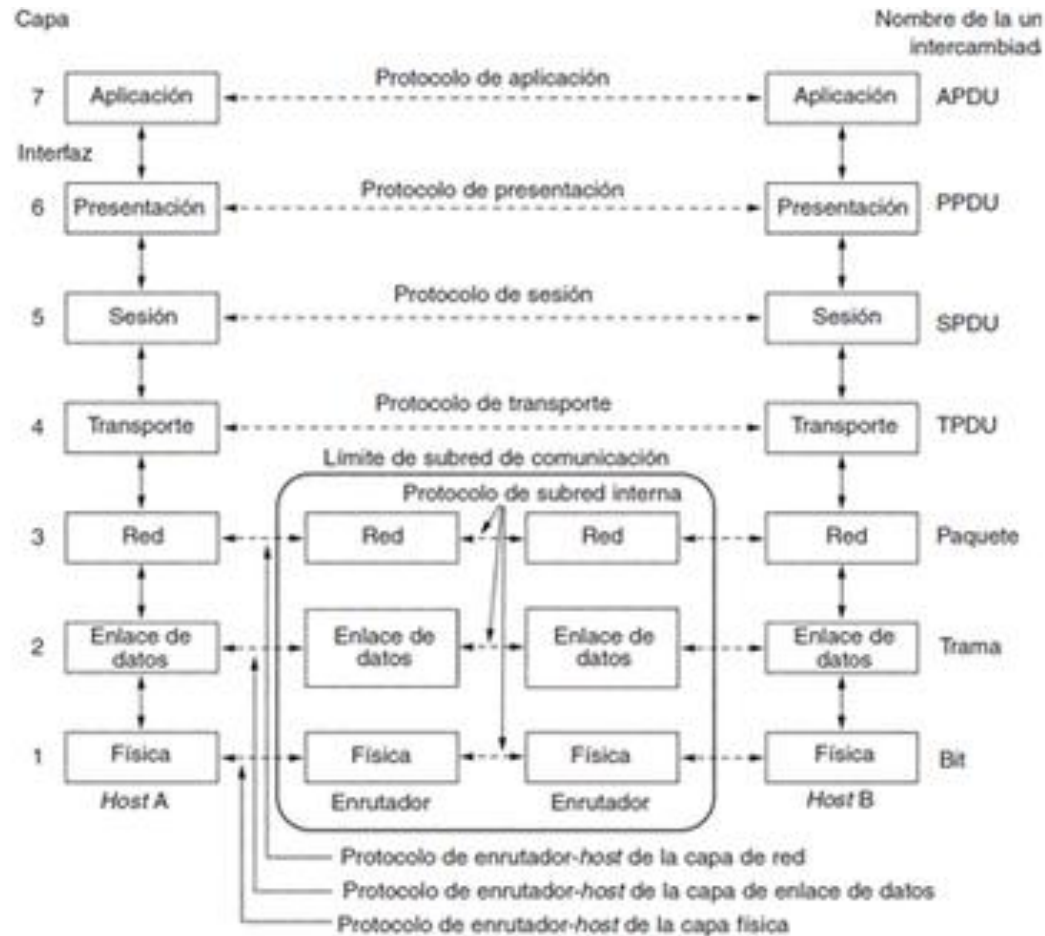
X.400, GIPSI (of INRIA) UIP  
X.400, RFC987 UIP

X.400, another UIP  
X.400, DFN UIP  
X.400, EARN/X.400 gateway UIP

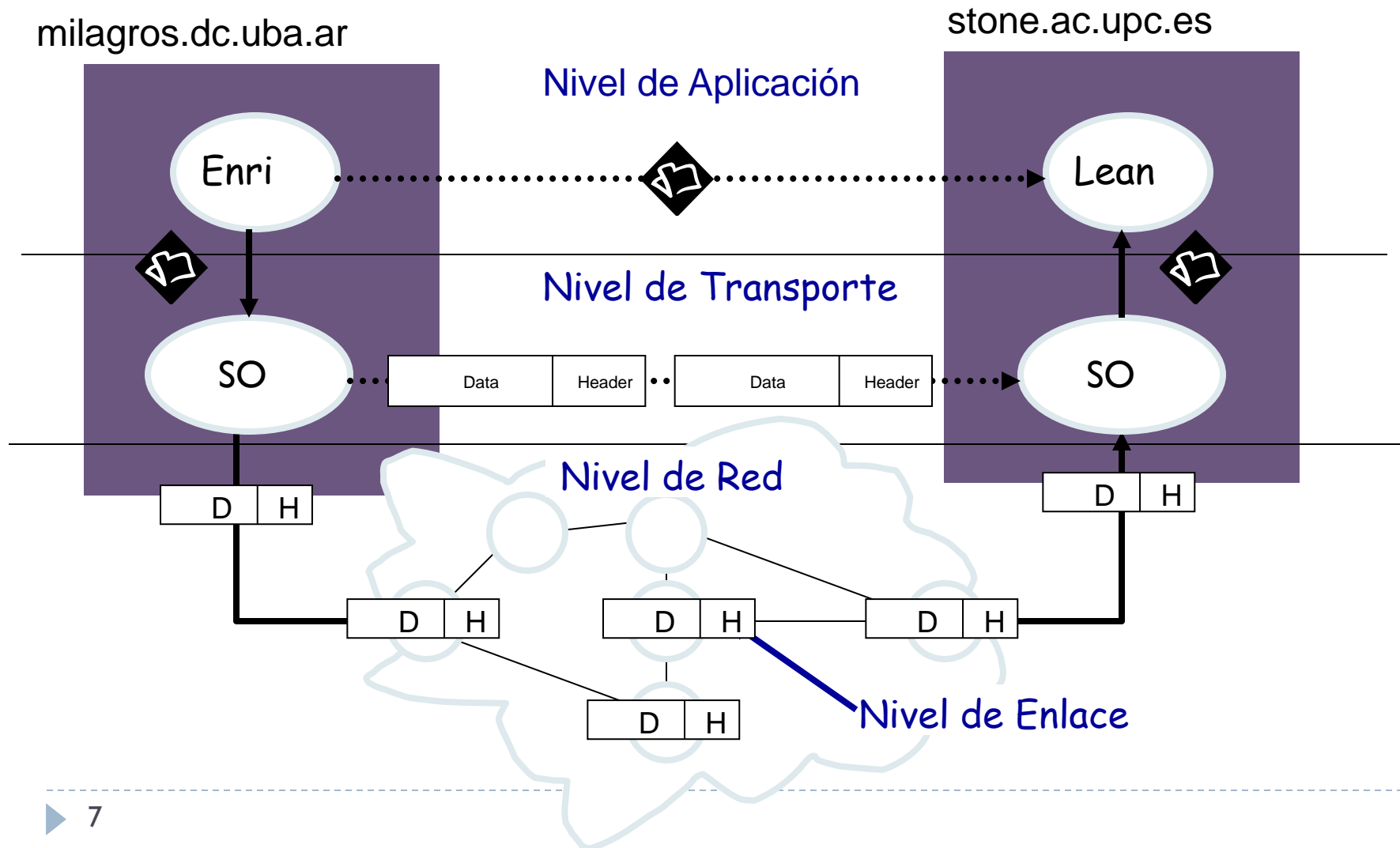
EAN, RFC822 UIP and domain order  
EAN, X.400 encoding, RFC987 UIP

**FIGURE 1. Sample Addresses for Different Networks**

# “El Modelo OSI”



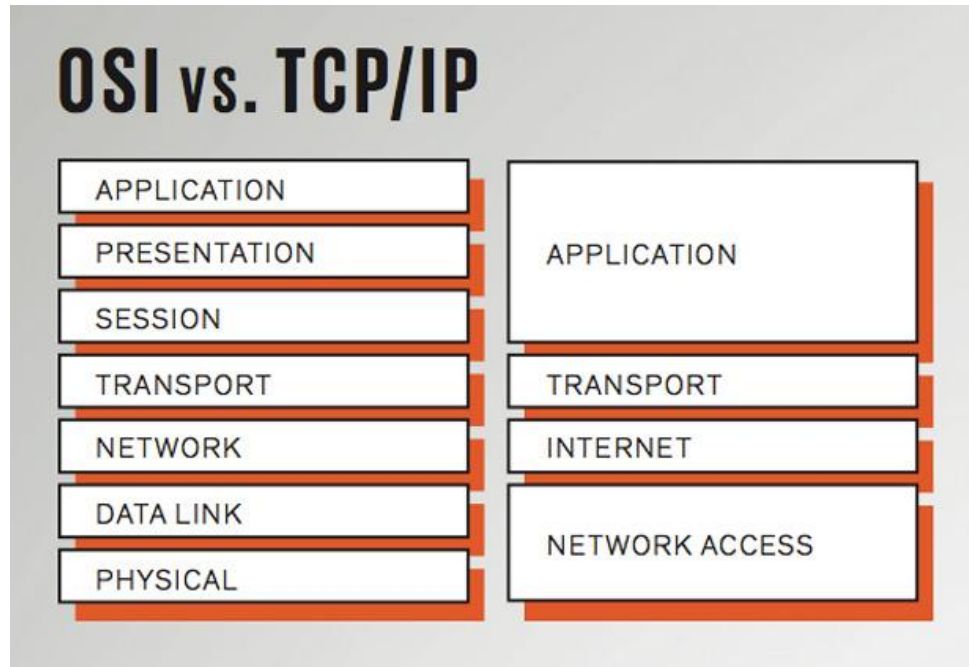
# Nivel Transporte : “End to End”



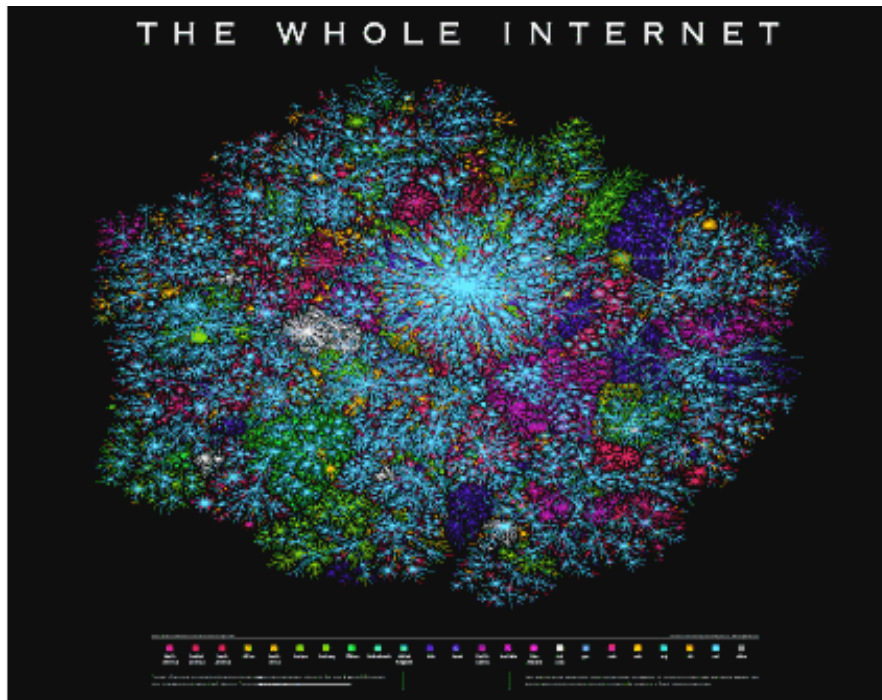
# Como se impuso TCP/IP a OSI ?

---

- ▶ Una visión de las causas la pueden encontrar en :
- ▶ <http://spectrum.ieee.org/computing/networks/osi-the-internet-that-wasnt>







# Nivel de Red

## Introducción

[http://www.?](http://www.?.)

# Agenda

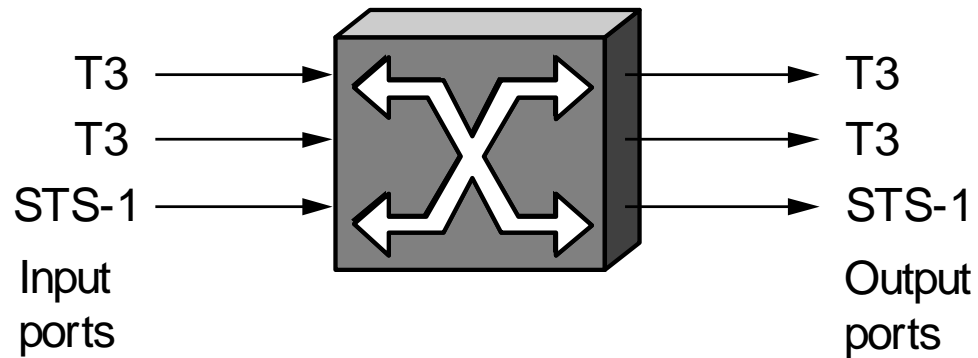
---

- ▶ Circuitos Virtuales
- ▶ Datagramas
- ▶ Introducción IP

# “Redes Escalables”

---

## ► Switch



# ¿Qué es un switch ?

---

- ▶ Es una “*appliance*” que interconecta enlaces para formar redes más grandes.
  - ▶ Un switch de datos es un dispositivo con múltiples entradas y múltiples salidas.
- ▶ Su trabajo es lograr que la mayor cantidad de paquetes que entren al switch vayan a la salida apropiada.
  - ▶ Envía paquetes, frames o celdas de un puerto de entrada a un puerto de salida (función conocida como *switching* ó *forwarding*)
  - ▶ El puerto de salida se selecciona utilizando una dirección que trae el header (encabezado) del paquete, frame o celda
- ▶ Según el tipo de switch:
  - ▶ Para distribuir los paquetes, algunos utilizan circuitos virtuales y otros conmutación de paquetes.
  - ▶ Pueden conmutar paquetes de longitud variable o de longitud fija.

# El switch permite construir redes escalables

---

- ▶ Los switches, al interconectarse unos con otros, permiten cubrir grandes áreas geográficas (además, toleran la latencia). Permiten construir grandes redes
- ▶ Pueden soportar un gran número de nodos (ancho de banda es escalable).
- ▶ Colocar un nuevo host al switch no necesariamente carga más la red

# Los dos grandes paradigmas...

Orientado a Conexión  
Sin Conexión

# Sin conexión : Datagramas

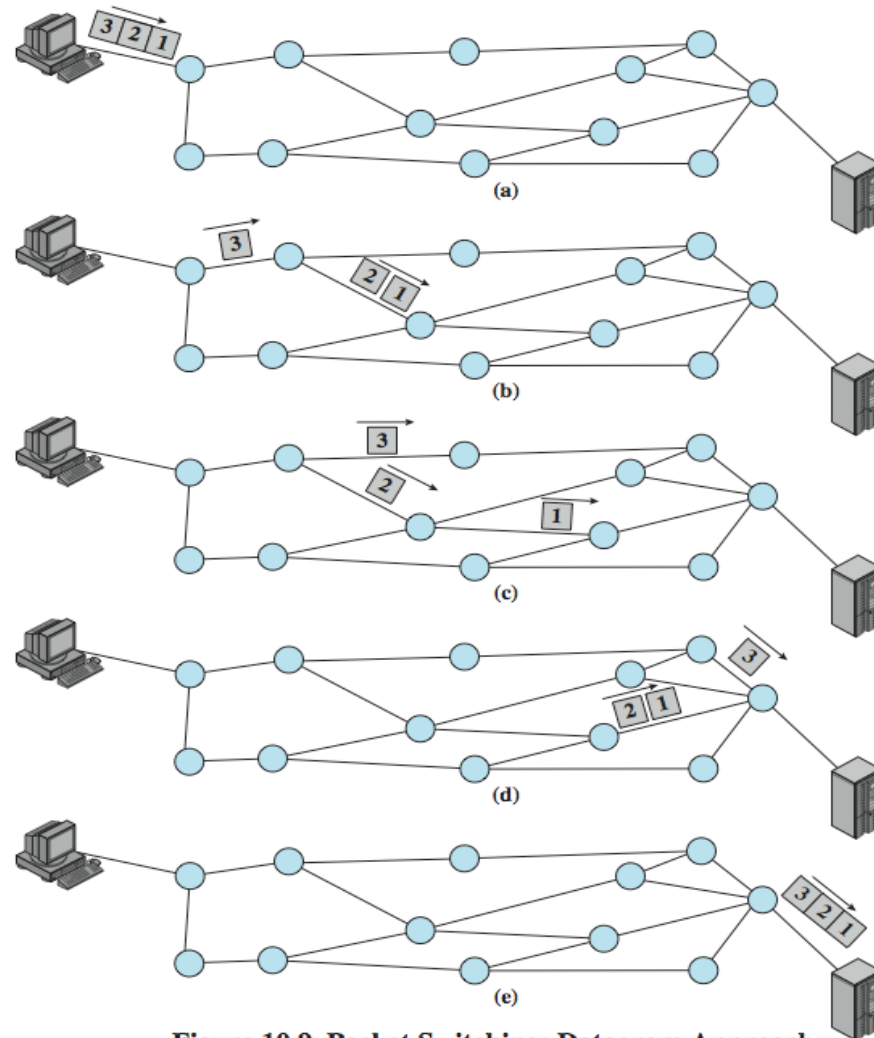


Figure 10.9 Packet Switching: Datagram Approach

# Orientado a Conexión: Circuito Virtual

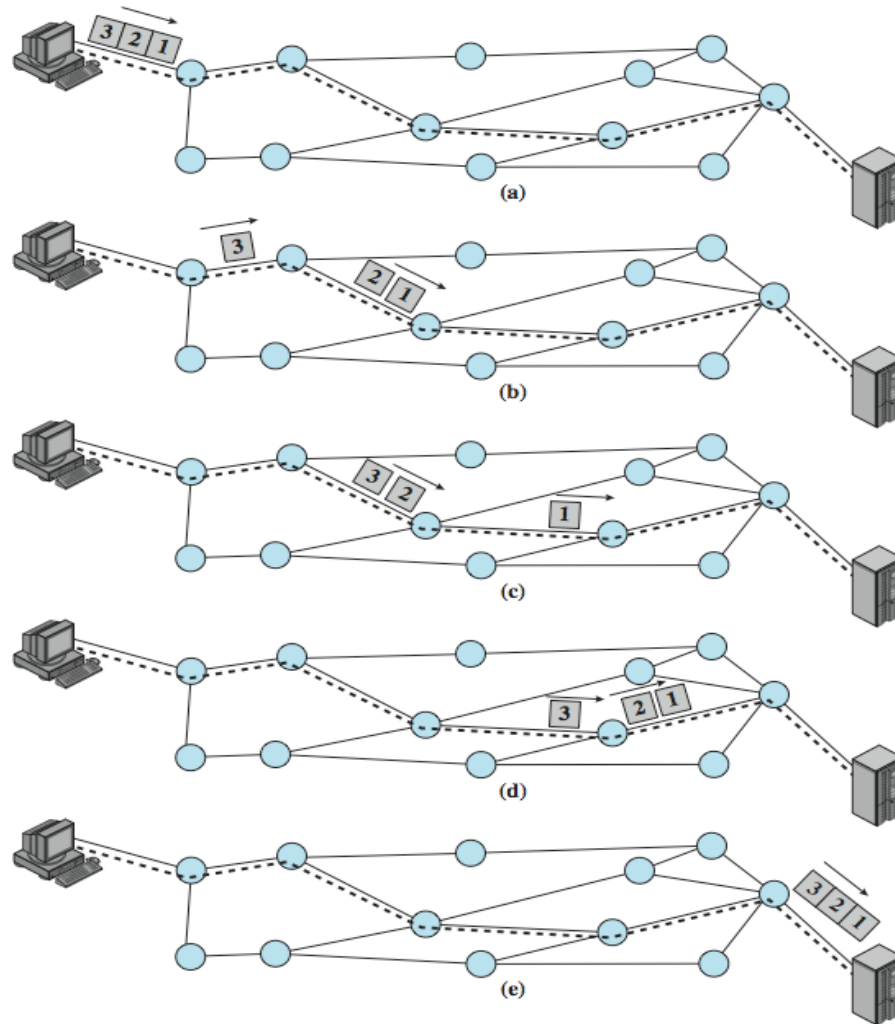


Figure 10.10 Packet Switching: Virtual-Circuit Approach

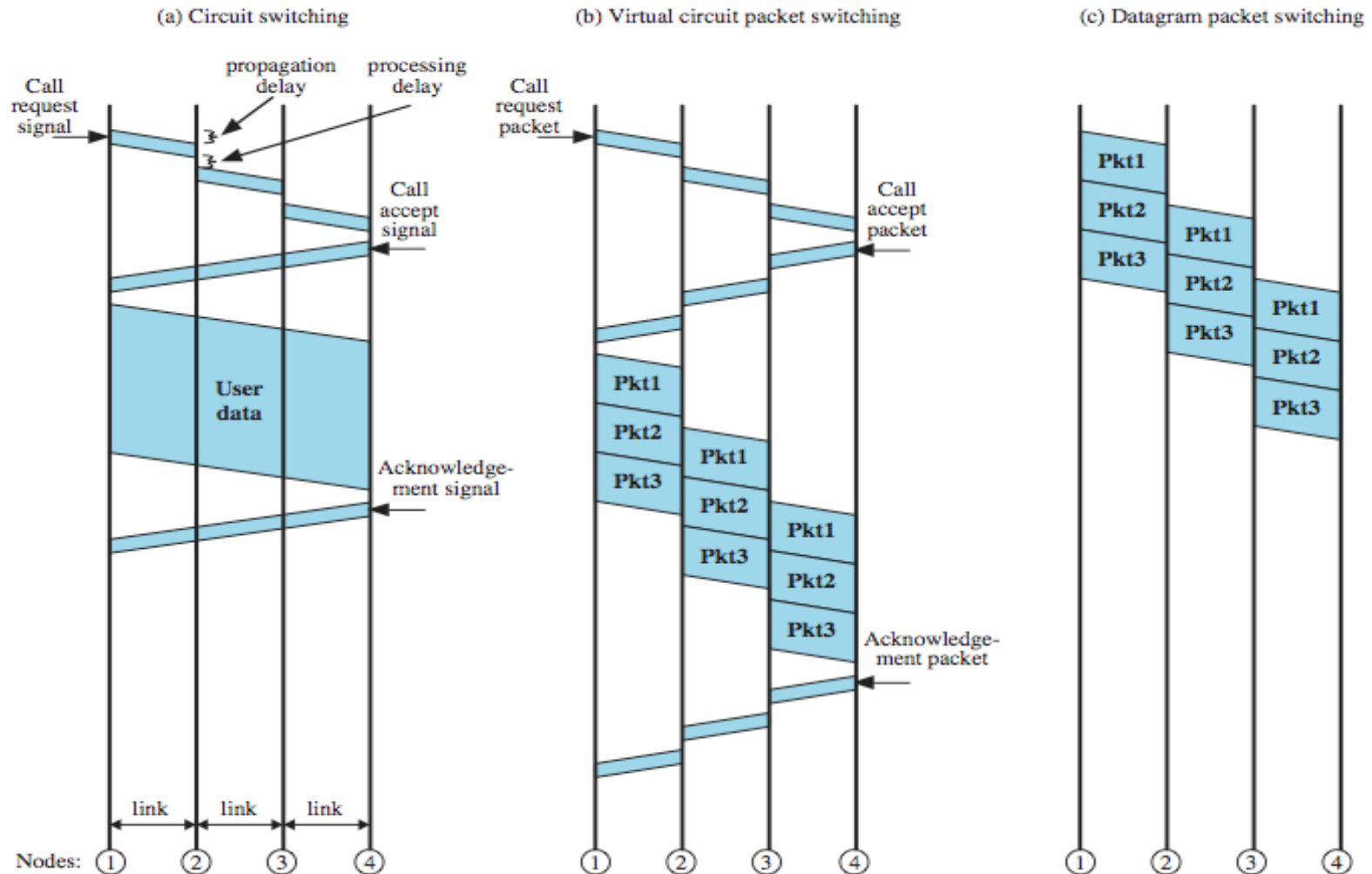


# Conmutación de circuitos vs conmutación de paquetes

---

- ▶ La performance depende de varios retardos:
  - ▶ Retardo de propagación
  - ▶ Tiempo de transmisión
  - ▶ Retardo de nodo
  
- ▶ También de otras características, incluyendo:
  - ▶ Transparencia
  - ▶ Overhead

# Temporización de eventos



# Conmutación no orientada a conexión (datagrama)

---

## ▶ Características

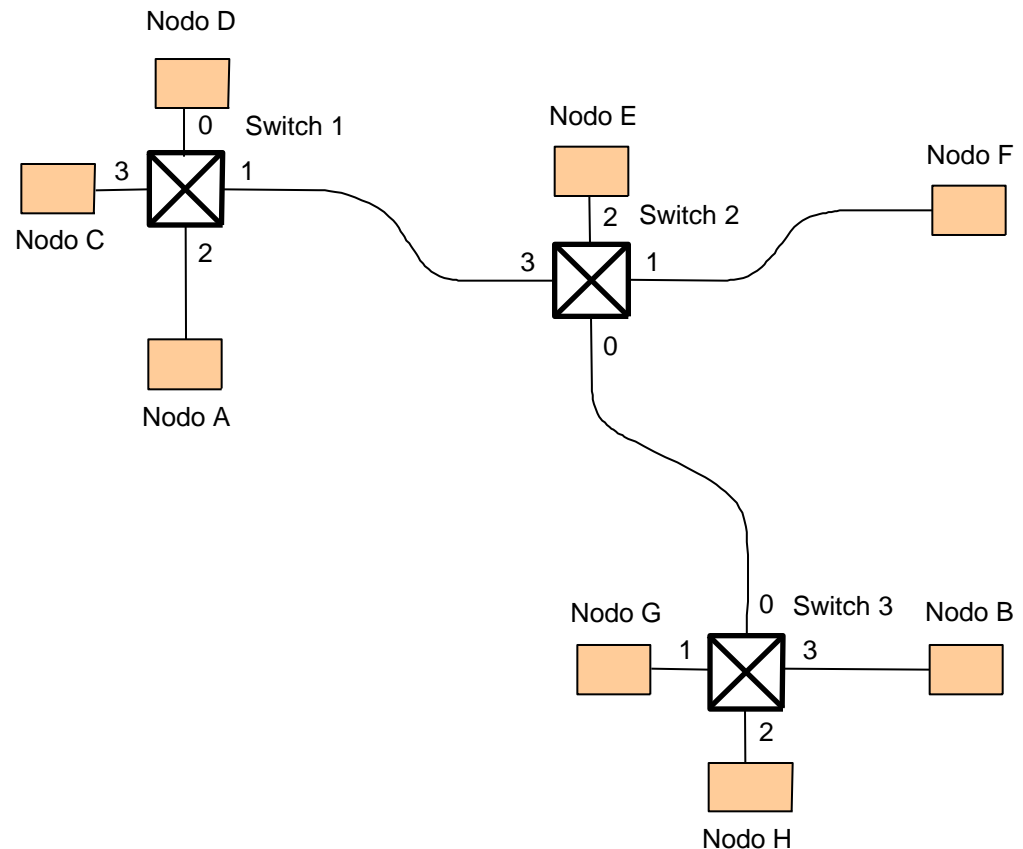
- ▶ No existe una fase para establecer una conexión
  - ▶ el nodo puede enviar el paquete *cuando quiera*.
- ▶ Cada paquete se envía independientemente y debe llevar toda la información necesaria para alcanzar su destino
- ▶ Llamado modelo *Connectionless* (no orientado a conexión) o de datagrama

# Conmutación sin conexión (datagrama)

Analogía: sistema postal

Cada switch mantiene una tabla de *forwarding* (*routing*)

Tabla de conmutación para el switch 2	
Destino	Puerto
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0



# Modelo de Datagrama

---

- ▶ No se debe esperar un RTT (*round trip time*) para establecer una conexión; un nodo puede enviar datos tan pronto como este listo.
- ▶ El nodo origen de los datos no tiene porque saber si la red es capaz de entregar un paquete o frame o si el nodo destino está listo para recibir los datos.
- ▶ Ya que los paquetes son tratados independientemente, es posible cambiar el camino para evitar los enlaces y los nodos que estén fallando.
- ▶ Ya que cada paquete lleva la dirección completa del nodo destino, la información adicional de control (*overhead*) que lleva es mucho mayor que la utilizada en el modelo orientado a conexión.

# Conmutación orientada a conexión (circuito virtual)

---

- ▶ Se requiere una fase para establecer una conexión y otra de finalización de la conexión
- ▶ Los paquetes o celdas que se transmiten después de establecer la conexión utilizan siempre el mismo circuito
- ▶ Llamado modelo *connection-oriented* (orientado a conexión) ó circuito virtual

# Conmutación orientada a conexión (circuito virtual)

Analogía: llamada telefónica

Cada switch mantiene una tabla VC

Tabla VC para el switch 1			
Puerto de entrada	VCI de entrada	Puerto de salida	VCI de salida
2	5	1	11

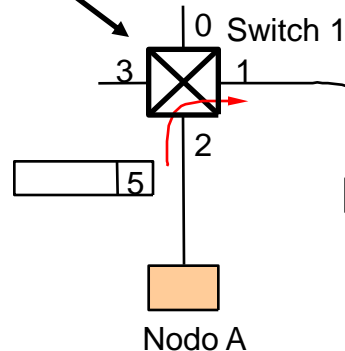


Tabla VC para el switch 2			
Puerto de entrada	VCI de entrada	Puerto de salida	VCI de salida
3	11	0	7

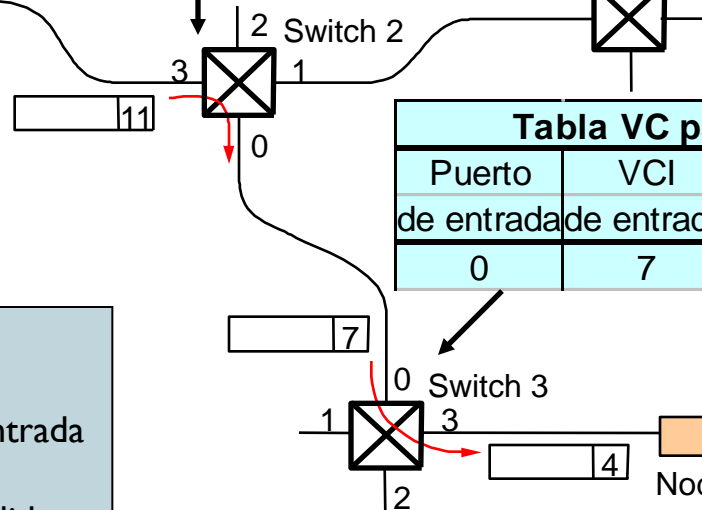


Tabla VC para el switch 3			
Puerto de entrada	VCI de entrada	Puerto de salida	VCI de salida
0	7	3	4

## Cada tabla de circuitos virtuales tiene:

1. El puerto por el cual llega el paquete.
2. El identificador del circuito virtual (VCI) de entrada
3. El puerto por el cual debe salir el paquete
4. El identificador del circuito virtual (VCI) de salida

# Tipos de conexiones

---

- ▶ **Conexión Permanente (PVC)**
  - ▶ Este tipo de conexión la define y la finaliza el administrador de la red: una persona solicita a la red la creación de los registros en las tablas VC. Después de creado el circuito virtual ya se pueden enviar datos.
- ▶ **Conexión por Solicitud ( o conmutado) (SVC)**
  - ▶ Cuando el nodo A desea enviar datos al nodo B envía un mensaje de solicitud de conexión a la red, luego el switch que la recibe se lo envía al siguiente, hasta llegar al nodo B. Este último, si acepta la conexión, devuelve el identificador de circuito que desea utilizar (4 en el ejemplo anterior) y esta “aceptación” se repite en todos los switches que se encuentran en el camino. Después de construir el circuito virtual se empieza a enviar datos.



# Finalización de la conexión

---

- ▶ **Conexión Permanente (PVC)**
  - ▶ El administrador de la red, una persona, solicita o hace las operaciones que permitan “bajar” el circuito virtual.
- ▶ **Conexión por Solicitud (SVC)**
  - ▶ Cuando el nodo A no desea enviar más datos al nodo B, termina el circuito virtual enviando un mensaje de finalización a la red. El switch que recibe el mensaje borra la línea de la tabla de VC correspondiente a ese circuito y envía un mensaje de finalización al siguiente switch para que repita la misma acción y así hasta alcanzar al nodo B. Si después de esto el nodo A envía un paquete o celda a la red, este puede ser descartado pues ya no existe el circuito virtual.

# Modelo de circuito virtual

---

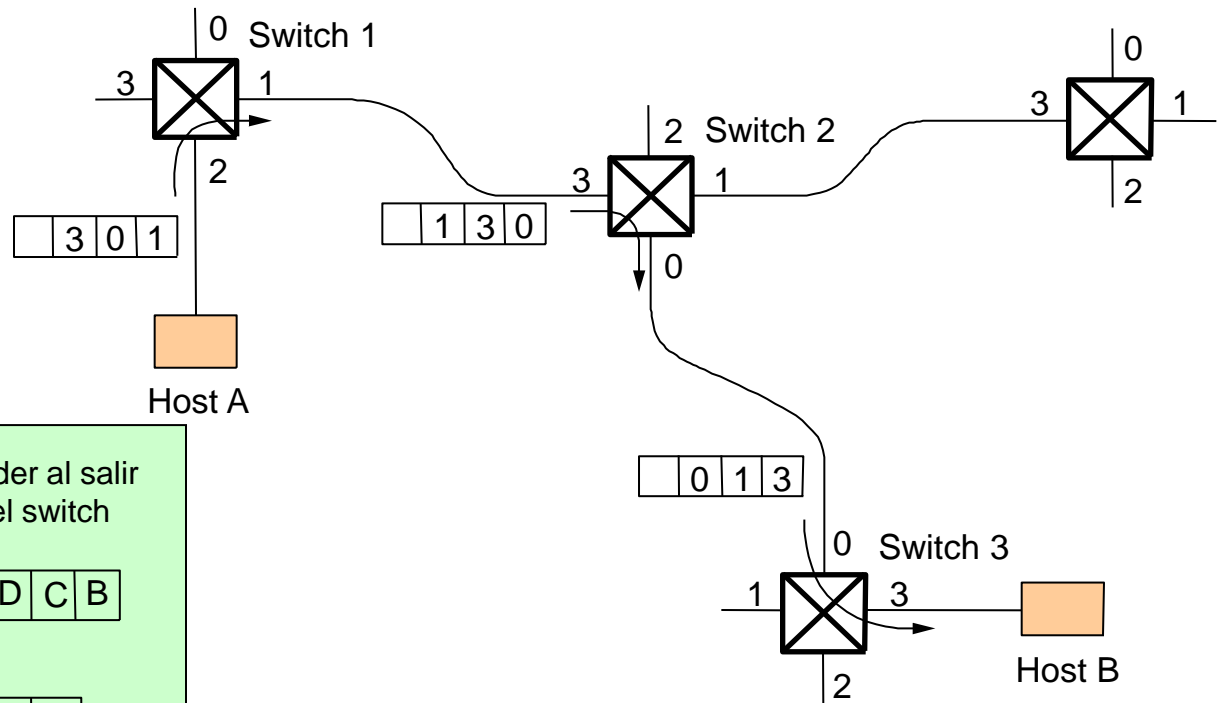
- ▶ Normalmente debe esperarse un RTT completo mientras se establece una conexión para poder enviar el primer paquete o celda.
- ▶ La solicitud de conexión debe llevar la dirección completa del nodo destino, pero los demás paquetes o celdas sólo tienen un identificador muy pequeño (el VCI) haciendo que el *overhead* sea pequeño.
- ▶ Si un switch o un enlace falla, el circuito virtual falla y una nueva conexión debe establecerse.
- ▶ Establecer una conexión de antemano, permite reservar recursos en los switches (espacio en buffers).
- ▶ “ Viejas “ Tecnologías que utilizan circuitos virtuales son: X.25, Frame Relay ( N2 ?) y ATM (N2/N3 ?).

# Conmutación *Source Routing*

---

- ▶ Toda la información sobre la topología de la red que se necesita para conmutar los paquetes es proporcionada por el nodo origen.
- ▶ Existen varias formas de implementar el Source Routing.
  - ▶ Rotación
  - ▶ Stripping
  - ▶ Pointer

# Conmutación *Source Routing*



# Uso de Source Routing

---

- ▶ La conmutación basada en *Source Routing* puede ser utilizada sobre redes no orientadas a conexión (datagrama) o en redes orientadas a conexión (circuito virtual). Por ejemplo:
  - ▶ IP (Internet Protocol), que es un protocolo no orientado a conexión, incluye una opción para *source routing* que permite que ciertos paquetes seleccionados para ser enrutados desde el origen.
  - ▶ En redes de circuitos virtuales, el *source routing* significa escoger un trayecto especificado sobre la red.

# Internetworking

---

Modelo de Servicio “Best Effort Service”

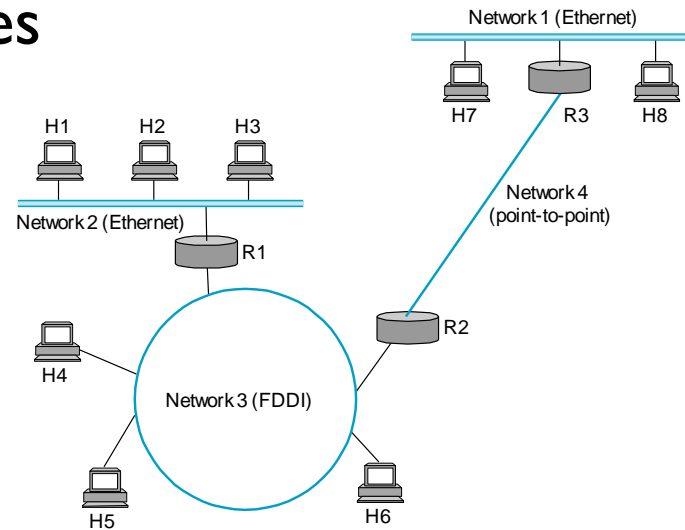
# Agenda

---

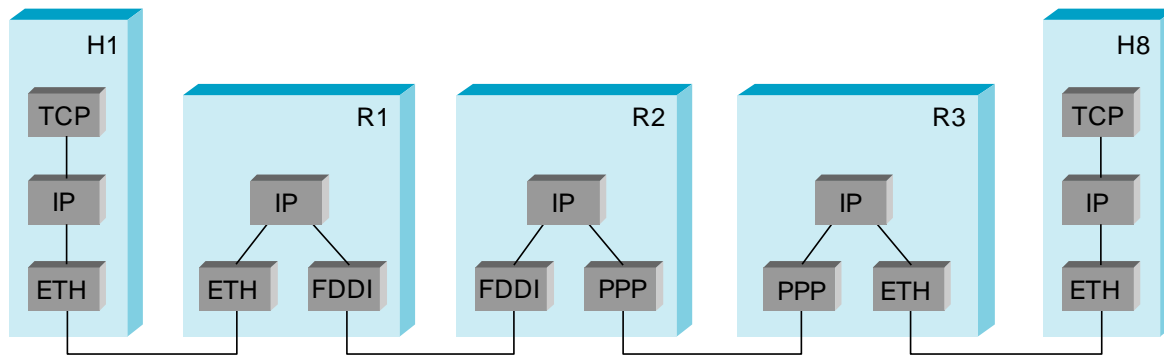
- ▶ Modelo de Servicio
- ▶ Cabecera IP
- ▶ Fragmentación
- ▶ Direccionamiento Global
- ▶ Forwarding

# IP Internet

## ► Interconexión de Redes



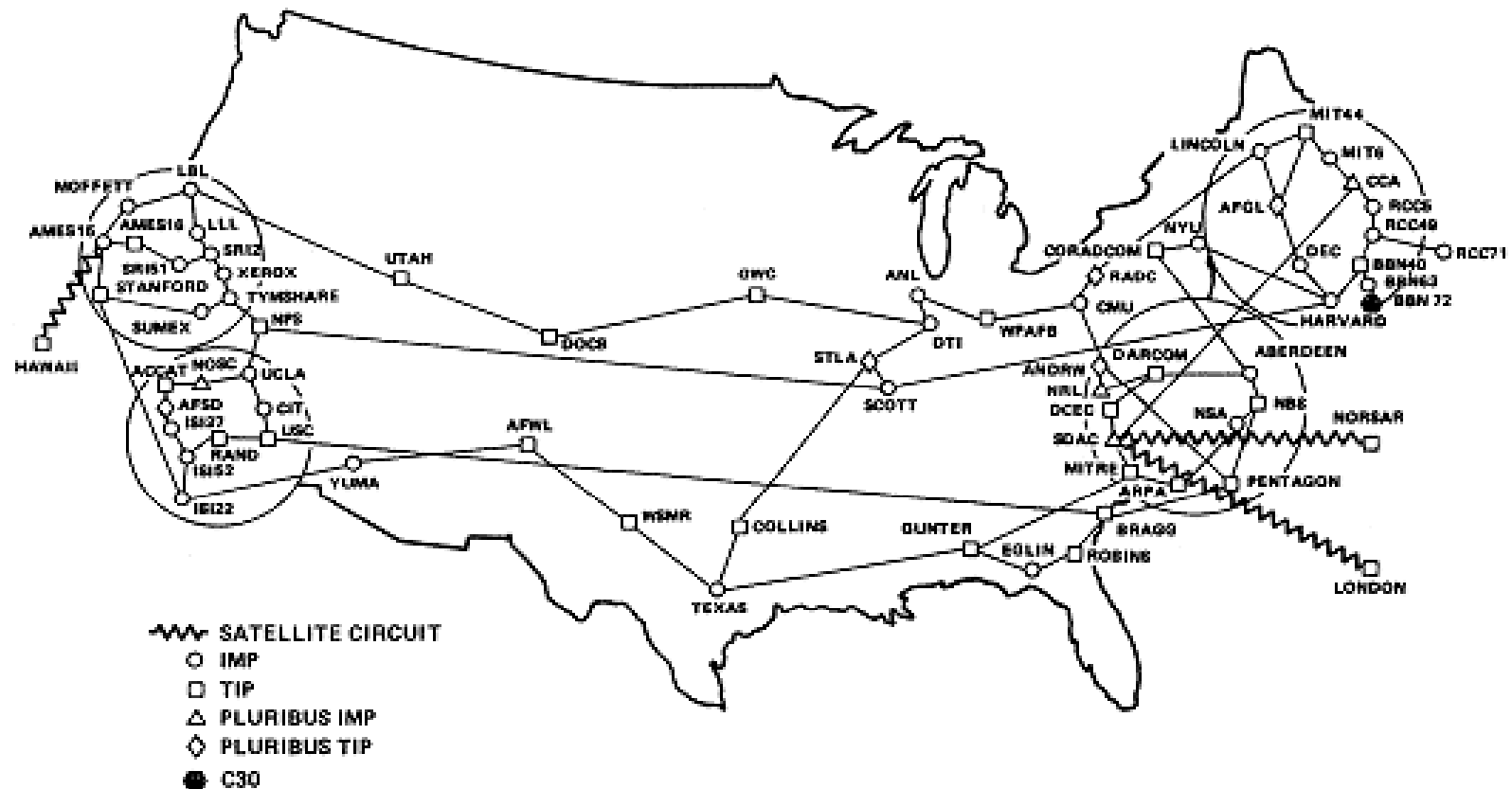
## ► Protocol Stack





# IP Internet

ARPANET GEOGRAPHIC MAP, OCTOBER 1980

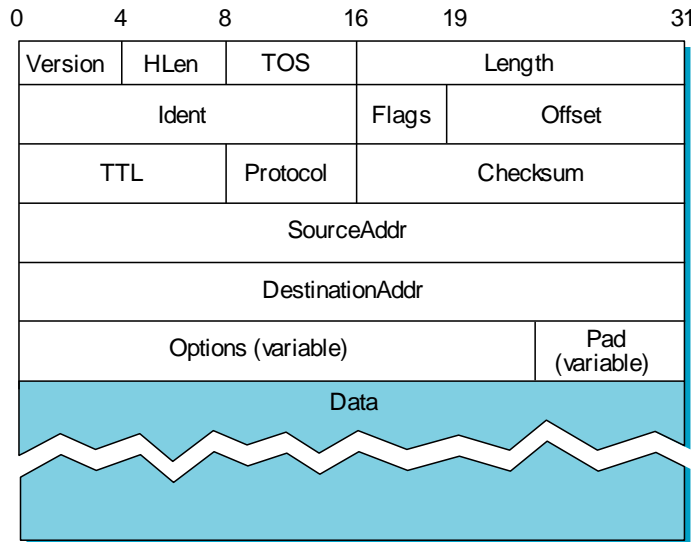


(NOTE: THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS)  
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

# Modelo de Servicio

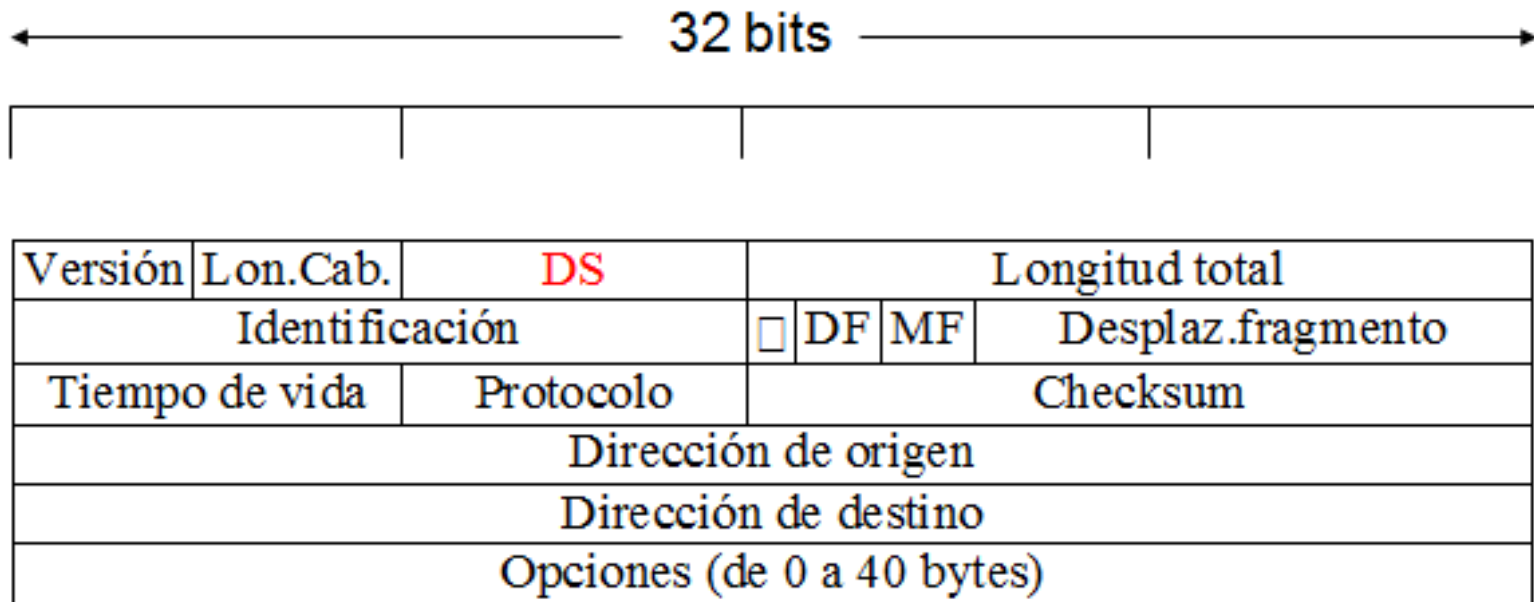
---

- ▶ Connectionless (datagram-based)
- ▶ Best-effort (unreliable service)
  - ▶ Paquetes se pueden perder
  - ▶ Enviar fuera de orden
  - ▶ Entrega de copias
  - ▶ No hay un cota para el tiempo de entrega
- ▶ Formato



# Cabecera IP ( versión 4)

---



La cabecera de un datagrama IP contiene información que deben interpretar los routers. El tamaño de la cabecera es normalmente de 20 bytes, pudiendo llegar a 60 si se utilizan los campos opcionales.

# Campos del header IP

---

- ▶ Versión: actualmente 4, comienzan los 6 pero el resto del formato del header no es el mismo en ambas versiones.
- ▶ Longitud Cabecera: en palabras de 32 bits (mínimo 5, máximo 15)
- ▶ Longitud total: en bytes, máximo 65535 (incluye la cabecera)
- ▶ Identificación, DF, MF, Desplaz.
- ▶ Fragmento: campos de fragmentación
- ▶ Tiempo de vida: contador de saltos hacia atrás (se descarta cuando es cero)
- ▶ Checksum: de toda la cabecera (no incluye los datos)
- ▶ Dirección fuente y destino 32 bits

# Algunos Valores de campo protocolo

Valor	Protocolo	Descripción
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP en IP (encapsulado)
5	ST	Stream
6	TCP	Transmission Control Protocol
8	EGP	Exterior Gateway Protocol
17	UDP	User Datagram Protocol
29	ISO-TP4	ISO Transport Protocol Clase 4
80	CLNP	Connectionless Network Protocol
88	IGRP	Internet Gateway Routing Protocol
89	OSPF	Open Shortest Path First

# IP Fragmentación y reensamblado

---

- ▶ Cada tecnología de red tiene a nivel de enlace un MTU (Maximum Transmission Unit)
  - ▶ Ethernet (1500 bytes), FDDI (4500 bytes)
- ▶ Estrategia : IP se adapta a la tecnología de red subyacente (MTU !!)
  - ▶ Fragmentación ocurre si un router recibe un datagrama que debe reenviar a una red donde su
$$\text{MTU} < \text{Tamaño\_datagrama}$$
  - ▶ Reensamblado se realiza en el host destino
  - ▶ Todos los fragmentos tienen el mismo identificador
  - ▶ Fragmentos son datagramas autocontenidos
  - ▶ IP no recupera fragmentos perdidos

# Fragmentación en IP

---

- ▶ Los fragmentos reciben la misma cabecera que el datagrama original salvo por los campos 'MF' y 'Desplazamiento del Fragmento'.
- ▶ Los fragmentos de un mismo datagrama se identifican por el campo 'Identificación'.
- ▶ Todos los fragmentos, menos el último, tienen a 1 el bit MF (More Fragments).
- ▶ La unidad básica de fragmentación es 8 bytes. Los datos se reparten en tantos fragmentos como haga falta, todos múltiplos de 8 bytes (salvo quizá el último).
- ▶ Toda red debe aceptar un MTU de al menos 68 bytes (60 de cabecera y 8 de datos). Recomendado 576

# Fragmentación

(a) Fugura (a) Paquete sin Fragmentar (b) Paquete fragmentado

(a)

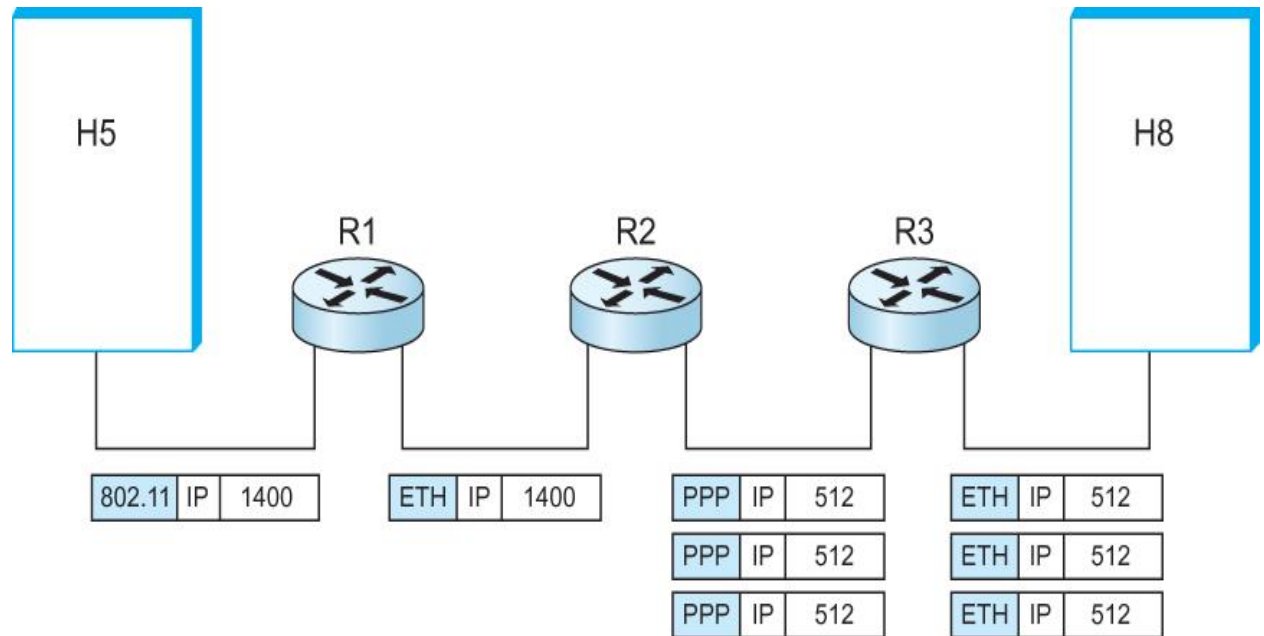
Start of header				
Ident = x			0	Offset = 0
Rest of header				
1400 data bytes				

(b)

Start of header				
Ident = x			1	Offset = 0
Rest of header				
512 data bytes				

Start of header				
Ident = x			1	Offset = 64
Rest of header				
512 data bytes				

Start of header				
Ident = x			0	Offset = 128
Rest of header				
376 data bytes				





# Direccionamiento Global

## ► Propiedades

- Globalmente única
- Jerárquica : red + host
- 4 Mil Millones de IP address, mitad clase A ,  $\frac{1}{4}$  son B, y un  $\frac{1}{8}$  son clase C

## ► Formato



## ► Notación “Dot”

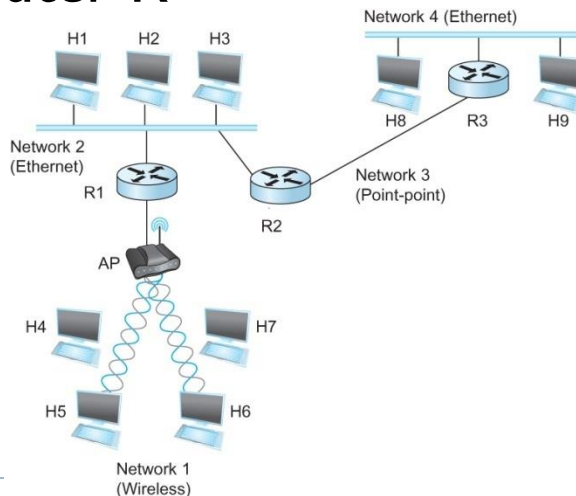
- 10.3.2.4
- 128.96.33.81
- 192.12.69.77

# IP Forwarding

## ► Estrategia

- cada datagrama tiene la dirección destino
- Si esta directamente conectado a la red destino => forward al host
- Si NO esta directamente conectado a la red destino => forward algún router
- forwarding “table maps network number” al “next hop”
- cada host tiene un “default router”
- cada router mantiene una “forwarding table”

## ► Para el router R



NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3

# IP Forwarding

---

## ▶ Algoritmo

```
if (NetworkNum del destino = NetworkNum de algunas de mis interfaces)
  then
    enviar datagrama al destino por esa interface
else
  if (NetworkNum del destino esta en mi forwarding table) then
    enviar datagrama al NextHop router
  else
    enviar datagrama al default router
```

En que caso el algoritmo de forwarding se reduce a :

```
if (NetworkNum del destino = a mi NetworkNum) then
  enviar datagrama al destino directamente
else
  enviar el datagrama al default router
```

# Material Adicional

X.25 ( únicamente por cuestiones afectivas)

ATM ( Histórico , genero un cuerpo de conocimiento muy grande , quedan aun muchos servicios)

# Circuitos virtuales v Datagramas

---

## ▶ Circuitos virtuales

- ▶ La red puede proveer control de secuenciamiento y de errores
- ▶ Los paquetes son enviados más rápidamente

## ▶ Datagramas

- ▶ No requiere fase de establecimiento
- ▶ Es más flexible



# X.25

---

- ▶ Norma de la ITU-T para la interface entre los host y la red de conmutación de paquetes
- ▶ Era Casi universal sobre redes de paquetes conmutadas
- ▶ Define tres capas:
  - ▶ Física
  - ▶ Enlace
  - ▶ Paquete



## X.25 - Física

---

- ▶ Interface en el enlace entre el nodo y la estación
- ▶ Los dos extremos son distintos:
  - ▶ Data Terminal Equipment DTE (user equipment)
  - ▶ Data Circuit-terminating Equipment DCE (node)
- ▶ La especificación de la capa física es: X.21
- ▶ Puede sustituir alternativa tal como la EIA-232



# X.25 - Enlace

---

- ▶ Link Access Protocol Balanced (LAPB):
  - ▶ Subset del HDLC
- ▶ Provee transferencia confiable de datos sobre el enlace
- ▶ Envía una secuencia de tramas





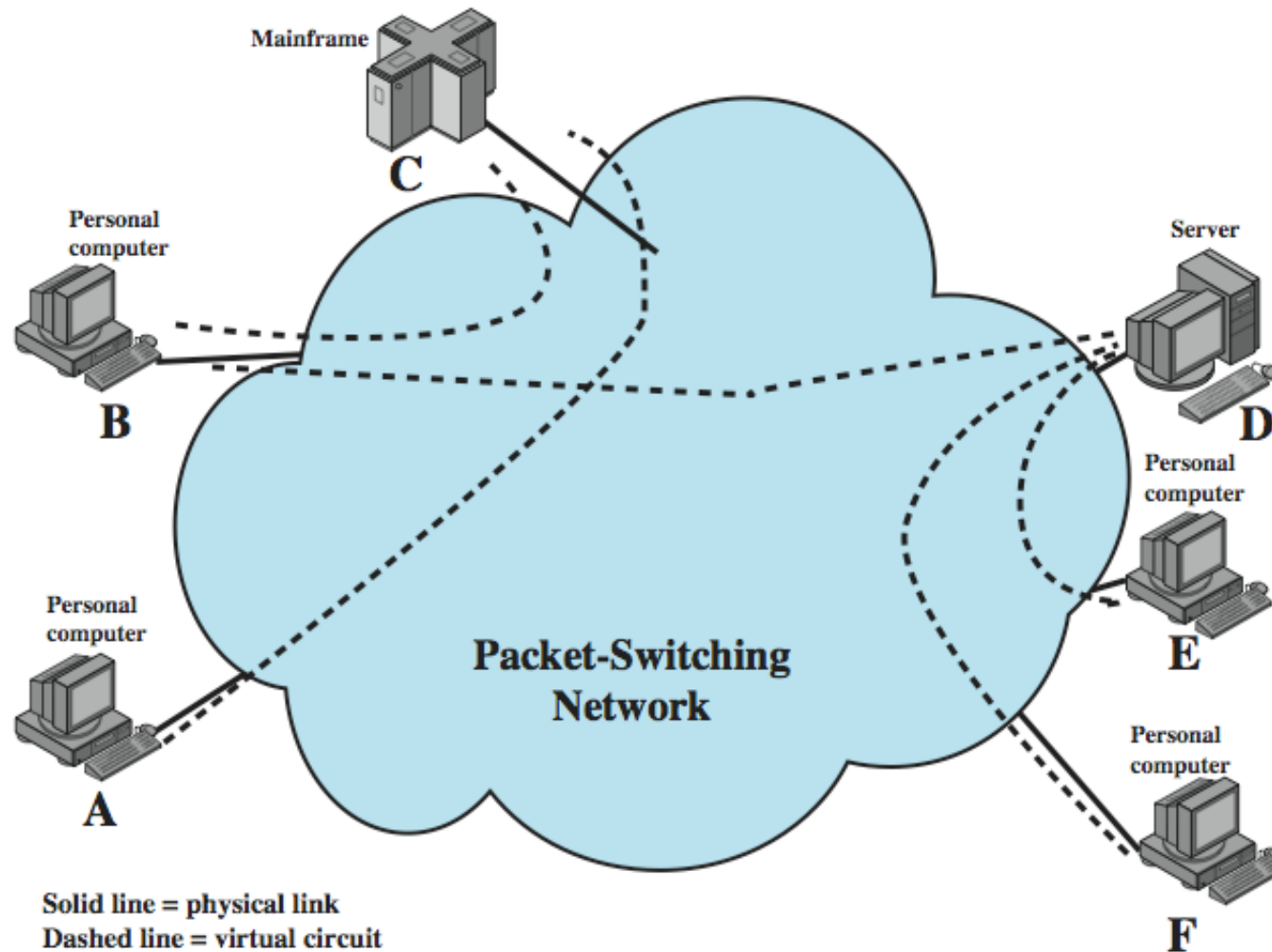
## X.25 - Paquete

---

- ▶ Provee conexiones lógicas (virtual circuits) entre los usuarios
- ▶ Todos los datos en esta conexión forman un solo tren entre las estaciones terminales
- ▶ Las conexiones son establecidas bajo demanda



# Uso de los circuitos virtuales en X.25



# X.25

---

- ▶ **Aspecto claves incluyen:**

- ▶ Paquetes de control de llamadas, señalización dentro de banda
- ▶ Multiplexaje de circuitos virtuales en el nivel 3
- ▶ Las capas 2 y 3 incluyen control de flujo y de errores



# ARPAC .....erases una vez ..

---

- ▶ “Recién en 1981 la empresa norteamericana que operaba las redes Telenet y Tymenet instalo un nodo que actuaba como sucursal bancaria en Buenos Aires. Informalmente se lo conocio como CIBA ya que se encontraba en la Central Internacional Buenos Aires. A finales del año siguiente ENTel, que era la unica empresa de telefonía con capitales puramente estatales, inauguro la primer red nacional de datos llamada ARPAC; esta red utilizaba el protocolo X.25 y quienes ganaron la licitacion para su dise~o y construccion fueron los mismos creadores de la red espa~ola IBERPAC. Por aquella epoca el panorama de la telefonía Argentina era lamentable, una linea telefonica para uso domiciliario costaba alrededor de 1200 dolares y el plazo de instalacion de la linea era imposible de saber. En algunos lugares ENTel tenia demandas de 15 y hasta 20 años de antigüedad porque no iniciaba la instalación de una linea. Para que tengan una idea mas clara aun de esta situacion, sucedia que una casa con las mismas características que otra valía entre 10.000 y 15.000 dolares mas solo por tener teléfono. La red nacional estaba basada en tres lineas troncales de cable coaxil y empalmes con redes de microondas con el resto del pais. Esa red telefonica era de una tecnologia anticuada y la unica red de datos (ARPAC), era de muy baja eficiencia, debido a los pocos nodos instalados solo en las ciudades principales y que obligaba al resto de los habitantes del pais a comunicarse por lineas analogicas hasta estos.

<http://www.govannom.org/e-zines/eko/eko03/eko03-12.txt>

---





ATM



# ATM: Generalidades

---

- ▶ Asynchronous Transfer Mode
- ▶ Frontera tecnológica en redes
- ▶ Utiliza celdas de longitud fija (53 bytes)
  - ◆ 5 bits de overhead
  - ◆ 48 bits de carga útil
- ▶ 155 Mbps y 622 Mbps



# ATM: Generalidades

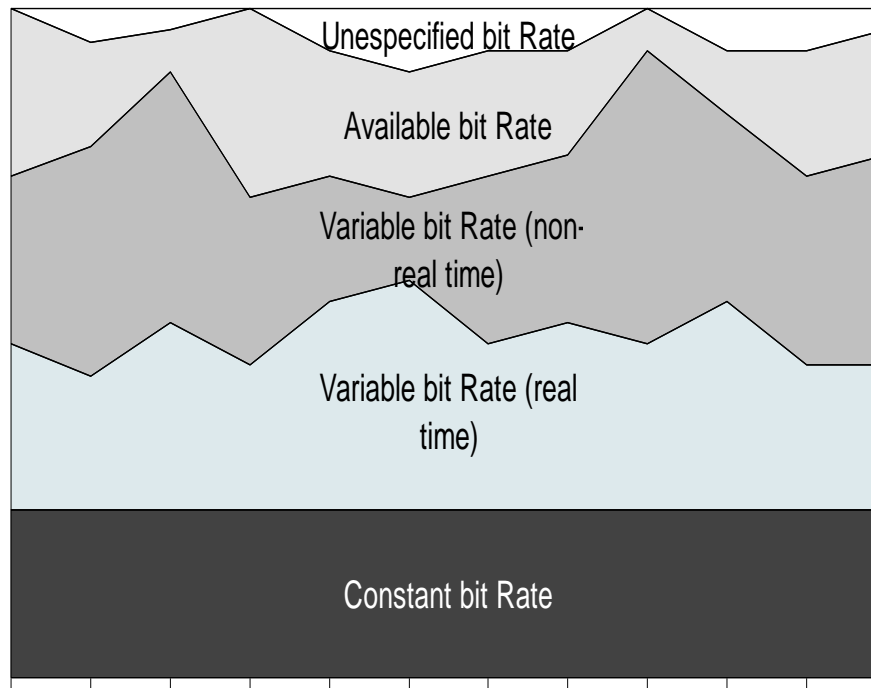
---

- ▶ Ofrece adaptación a distintas clases de servicios
- ▶ Se implementa con conmutadores de hardware y multiplexores de altísima capacidad
- ▶ Circuitos virtuales:
  - ▶ permanentes
  - ▶ conmutados



# Caracterización del tráfico

---





# Comparación con otras redes

---

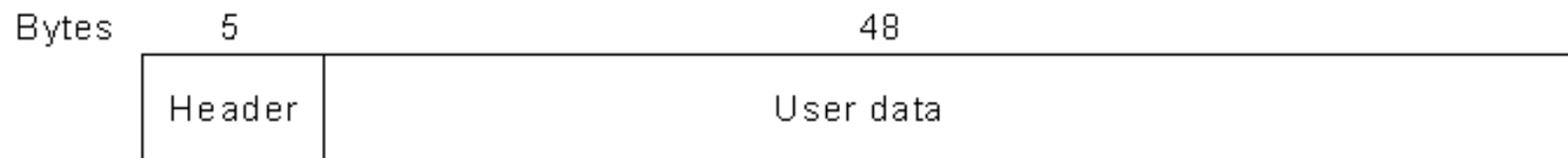
	Conmutación de Circuitos	Conmutación de paquetes (X.25)	Relé de tramas (Frame Relay)	ATM
Multiplexación	TDM	estadística	estadística	estadística
Optimización del uso por los puertos	no	si	si	si
Alto throughput	si	no	si	si
Retardo	muy bajo	elevado	bajo	muy bajo
Recuperación de errores por la red	no	si	no	no
Longitud de los paquetes	no se aplica	variable	variable	fija
Aplicación	red de transporte	servicio de inter-conexión de datos	servicio de inter-conexión de datos	red de transporte



Característica	X.25	Frame Relay	SMDS	ATM
Estándares	UIT-T, ISO	UIT-T, ANSI, IEEE	Bellcore, ANSI	UIT-T, ATM FORUM
Velocidades actuales	9,6 a 64 kbps (hasta 2 Mbps)	56 Kbps a 2048 Kbps	1,5 Mbps a 34 / 45 Mbps	155 Mbps a 2.5 Gbps
Longitud del paquete	Variable, hasta 4096 bytes	Variable, hasta 4096 bytes	Variable, hasta 9188 bytes	Fija, 53 bytes
Multidireccionamiento	No	Si (poco implementado)	Si	Propuesto
Direccionamiento	X.121, de longitud variable (>= 14 dig)	Fijo (DLCI de 10 bits)	Variable (10 a 15 dig. basados en nros. telef.)	Fijo (24 bits de VPI / VCI)
Servicio sin conexión	No	No	Si	No
Circuito virtual permanente	Si	Si	No se aplica	Si
Circuito virtual conmutado	Si	Si	No se aplica	Si
Control de flujo explícito	Si	No	No se aplica	No
Corrección de errores a nivel de enlace	Si	No	Si	No

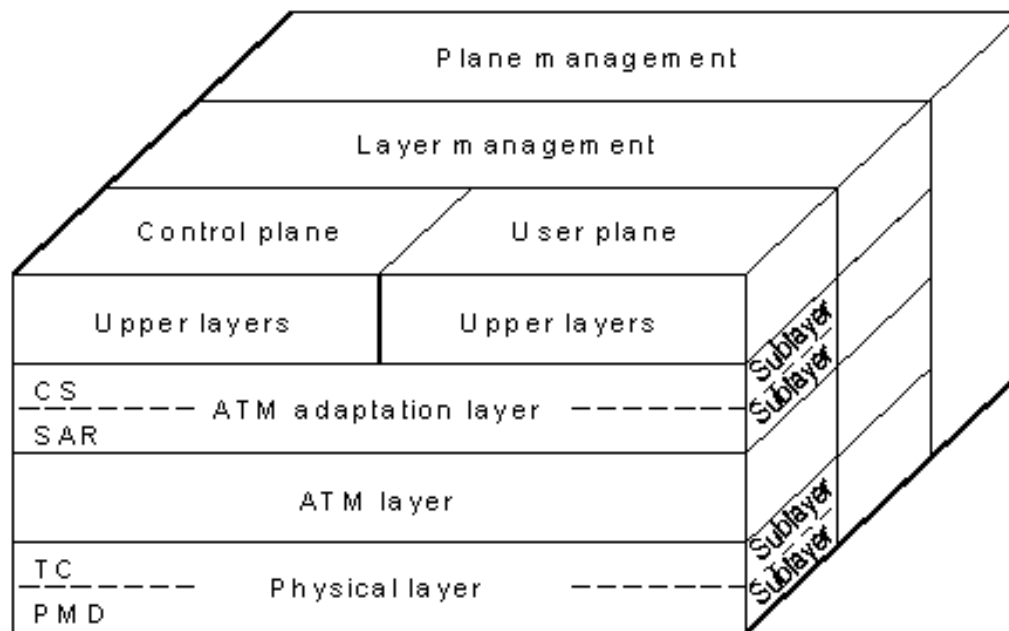
# Celda ATM

---



# El modelo B-ISDN ATM

---



CS: Convergence sublayer  
SAR: Segmentation and reassembly sublayer  
TC: Transmission convergence sublayer  
PMD: Physical medium dependent sublayer

# Niveles ATM y sus funciones

---

OSI layer	ATM layer	ATM sublayer	Functionality
3/4	AAL	CS	Providing the standard interface (convergence)
		SAR	Segmentation and reassembly
2/3	ATM		Flow control Cell header generation/extraction Virtual circuit/path management Cell multiplexing/demultiplexing
2	Physical	TC	Cell rate decoupling Header checksum generation and verification Cell generation Packing/unpacking cells from the enclosing envelope Frame generation
1		PMD	Bit timing Physical network access

